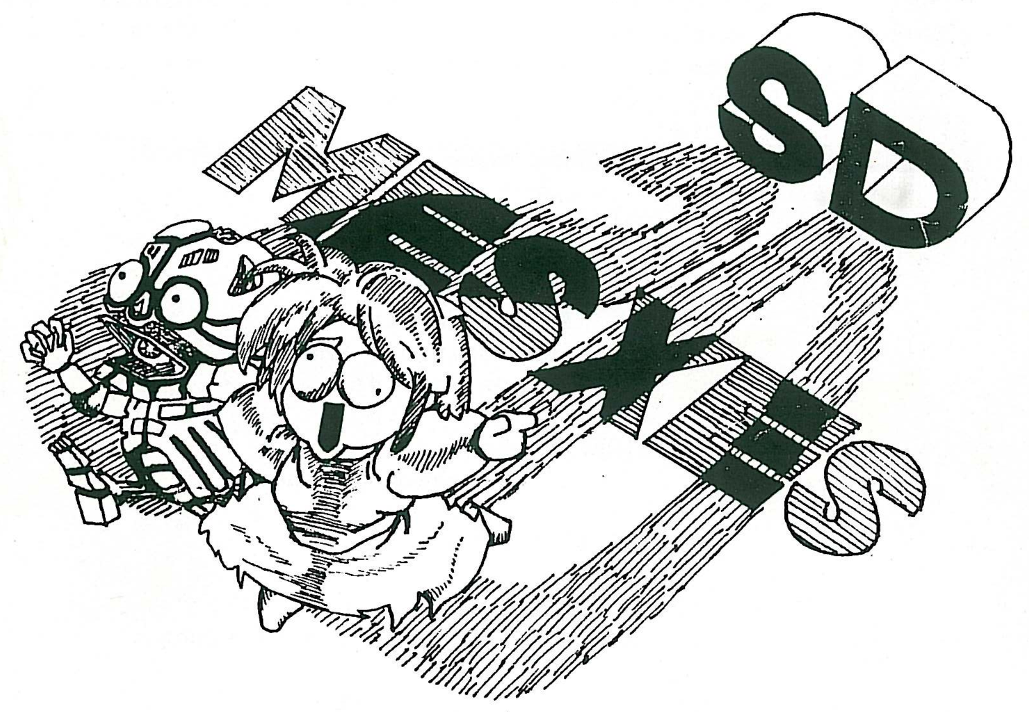
SD MESXES

Numero 1 Agosto 1.994



S.PICTUREDISK#12
V REUNION DE
USUARIOS DE MSX
NOTICIAS

ENSAMBLADOR
BLACK CYCLON
y mucho mas ...

250 ptas.

SUMARIO

SD MESXES número 1 / Agosto de 1.994

EDITA: Club Mesxes

REDACCION: Ramón Serna
Néstor Soriano
Juan Salvador Sánchez
Marcos Rosales

MAQUETACION: Ramón Serna
J. Salvador Sánchez

FOTOGRAFIA: Ramón Serna

COLABORAN: Mr. Henrik Gilvad, OASIS, Manuel Pazos, Enrique Martinez.

DIRECCION: Club Mesxes
C/ Manacor 16, 1º 1ª
07006 - Palma de Mallorca
Baleares

TELEFONO: (971) 46 17 13 : Ramón Serna (hijo)

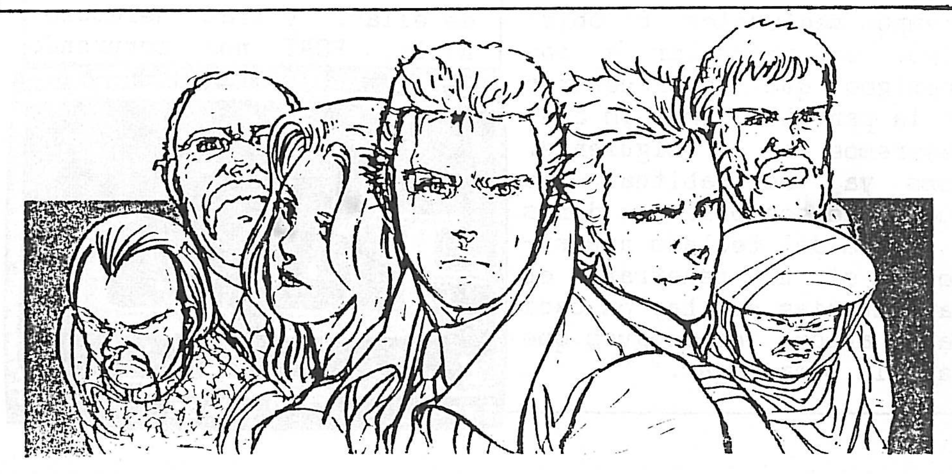
INTRODUCCION

Ha llegado el fanzine que estabas esperando. Totalmente dedicado al mundo del MSX. Trataremos los temas de mayor interes para los usuarios, novedades nacionales e internacionales, japonesas, holandesas, y del resto del mundo, con articulos claros pero completos, grandes ideas y buenos proyectos.

SD MESXES puede adquirirse unicamente por correo, al precio de 200 pts, con los gastos de envio incluidos, para los contrarrembolsos y certificados, consultad a la redacción.

Cualquier colaboración o ayuda al club será bien recibida y por supuesto gratificada. Para contactar con nosotros, encontrareis la dirección en la página anterior.

Ya sabemos que estais ansiosos por abrir el fanzine y devorarlo con los ojos ,¿o no?, pero antes debemos informaros que este fanzine no es como los que llevais viendo hasta ahora, sino que es un nuevo proyecto para la comunicación de las masa asi que: Ten cuidado hermano de no tener en el culo lo que tienes en la mano ,o , lo que es lo mismo: Pasa la página ya coño! J. Salvador Sanchez / Ramón Serna



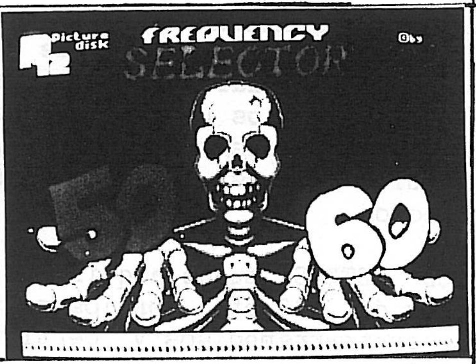
S. PICTUREDISK

De nuevo nos sorprende SUNRISE con el S. PICTURE DISK#12, el último hasta el momento...

Para empezar, se nos presenta el inevitable selector de frecuencia, realizado esta vez por CAIN. Sin duda este es un buen principio, con buenos gráficos y una música, en stereo (por supuesto), que no esta nada mal.

Tras la selección de Hz. nos encontramos con la demo-promo de BLADE LORDS, el último juego de PARALLAX, quien como ya sabreis es el responsable de juegos como el BLACK CICLON entre otros. Esta demo nos servirá de menú para acceder a los diferentes apartados.

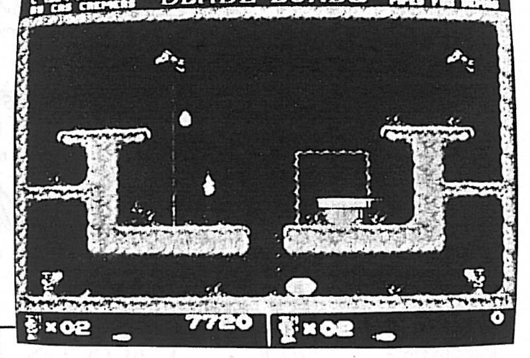
El juego, cuya música no oireis de no poseer un FM-PAC, es de tipo plataformas, algo asi como el BUBBLE BOBBLE, aunque ambientado en tiempos medievales. El objetivo, es aniquilar a los enemigos que se encuentran en la pantalla, tras lo cual pasaremos a la siguiente. Como ya es habitual, se pueden seleccionar las demos a través del teclado numérico, o si no, penetrando en las tuberias que hay en cada fase, aunque os aseguro que vale la pena jugar.



Sobre BLADE LORDS, solo me queda decir que el juego "de verdad" consta de 80 fases, y al igual que en la promo pueden jugar dos players a la vez.

A continuación, os resumo brevemente el contenido de las demos incluidas en el disco:

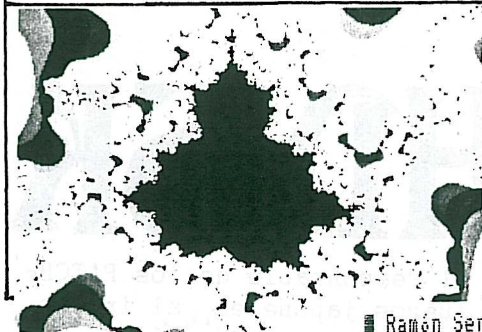
COLORWAVE: Es la primera de ellas, y tras seleccionarla, FONY nos sorprende



con un simulacro de reset. Una original forma de empezar...

Tras esto, y despues de un largo scroll vertical nos encontramos con el plato fuerte: PLASMAS... Si, y no hay uno, ni dos, por cada tecla del ordenador que pulsemos nos saldrá uno diferente. Además, con SHIFT podemos cambiar el RGB.

IMPACT BBS: El principio de esta demo decepciona un poco, pero así como avanza va mejorando. Lo principal de la demo son los tres fractales que se nos muestran (pueden cambiarse con 1,2 y 3). Los colores son ajustables, al igual que la velocidad y además con las teclas de funcion cambiamos el scroll. Por tanto, vale la pena experimentar con el teclado.



SCREEN 11 DESIGNER: No hay mucho que decir, puesto que se trata de un editor gráfico en screen 11.

PIXEL PARTY: Como de costumbre aquí se encuentra la "art galery" de SUNRISE, aunque esta vez con UN solo dibujo..

NEWS & INFO: En este apartado Sunrise informa sobre las novedades en torno al MSX, pudiendo elegir como siempre entre ingles o holandes, todo un detalle!

Y hasta aquí llega el SUNRISE PICTURE DISK#12. Hay que decir que cada vez son mejores y yo ya estoy ansioso por tener el número 13.

Ramón Serna / Néstor Soriano / J. Salvador Sánchez

STREET FIGHTER

A pesar de ser un juego original de maquina recreativa, siendo las posteriores versiones para consola, no podia faltar, para sorpresa de muchos, el STREET FIGHTER, para MSX.

Seria una total perdida de tiempo intentar comentar este juego, porque todos sabemos perfectamente de que va, y posiblemente todos hayamos echado alguna partidilla en el salón recreativo.

Pués bién, este juego, o mejor dicho, la promo, se encuentra comprimido bajo el nombre de SF.PMA en el SUNRISE MAGAZINE#13.

Cuenta unicamente de 3 personaje, y solo hay una ronda, si ganas sigues y si pierdes no. Tampoco es la replica exacta del SF para maquina, pero consta de muchos movimientos, y la velocidad es muy aceptable.

Lamentablemente, en el S.M.#13 hay un texto explicando el juego, pero está en Holandés.

Ramón Serna

PATCHDISK

Como todos sabeis, OASIS es el responsable de los PATCHDISK, discos traductores de juegos japoneses al ingles. Hasta ahora pueden encontrarse estos:

SD-SNATCHER - English patchdisk
XAK II - " " " "
XAK III - " " " "
YS II - " " " "
YS III - " " " "
YS Player's - Guidebook (manual)
RUNEMASTER 2- English patchdisk

Más información: RAMON RIBAS CASASAYAS, C/Sardenya 379 Atic 3a, 08025 - BARCELONA.

OASIS / Ramón Serna

PINK SOX MANIA

Son casi infinitos los trucos que muchos usuarios inventan para poder "observar" los Pink Sox Mania con su paleta de colores y sin censura...

A continuación os incluimos un pequeño programa con el que podreis transformar el modo de almacenar la paleta de la mayoria de PINK SOX MANIA, al igual que SAURUS LUCH, y muchas más imagenes. Así, ademas de conseguir cargar las imagenes con un simple BLOAD, también os librareis de esos molestos montones de pixels que se situan justamente donde no deberian estar.

```
10 SCREEN 7:COLOR ,0,0:C=0
20 READ N$:IF N$="!" THEN END
30 BLOAD N$+".sr7",S
40 OPEN N$+".pl7" AS#1 LEN=1:FIELD#1,1 AS A$
50 FOR K=1 TO 32 STEP 2
60 GET#1,K:A=ASC(A$)
70 R=A\16:B=AMOD16
80 GET#1,K+1:A=ASC(A$)
90 G=A
100 COLOR=(C,R,G,B)
110 C=C+1
120 NEXT K
130 BSAVE N$+".pl7",&HFA80,&HFAA0,S
140 C=0:CLOSE#1:GOTO 20
1000 DATA xxxxxxxx,xxxxxxxx,xxxxxxxx,... ' ficheros sin
1010 DATA xxxxxxxx,xxxxxxxx,xxxxxxxx,... '  extensión
```



Una vez ejecutado este programa podreis cargar las imágenes con paleta y sin estorbos haciendo simplemente:

```
10 SCREEN7
20 BLOAD"xxxxxxxx.pl7",S:COLOR=RESTORE
30 BLOAD"xxxxxxxx.sr7",S
40 GOTO 40
```

Ahora, está en vuestras manos el haceros un cargador más o menos decente o no.

Néstor Soriano / Ramón Serna

V REUNION DE

Como todos sabeis, dia 8 de Mayo del presente año se celebró en BArcelona la V Reunión de Usuarios de MSX. Vamos a intentar contaros todo lo que pasó, en el mismo orden en que lo hicieron allí.

El montaje fue bastante rápido, el equipo con el que se contaba era de un MSX TURBO R (gt), un televisor bastante grande y una mini-cadena de la que se aprovechaban los altavoces. El sonido, a pesar de tener un Music Module no era stereo por problemas de conexión.

Los primeros en mostrar algo fueron los de FKD, que abrieron la sesión con la presetación del disco duro, o más concretamente el interface (SCSI), para MSX. Se trataba del modelo antiguo, aunque de todas formas pudimos observar una maravilla. Especialmente cuando pusieron el Saurus Lunch y otras demos (...) en las que se podia escuchar la música de MoonBlaster al mismo tiempo y sin relentizar que el disco duro iba cargando imágenes.

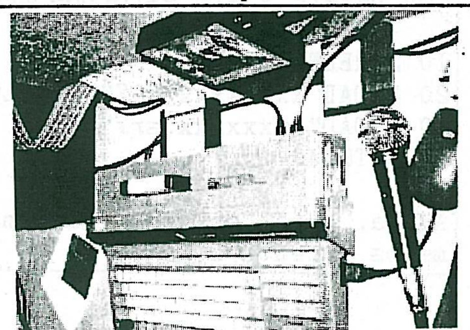
Al acabar dicha demostración, Ramón Ribas, organizador de la reunión, nos hizo un breve discurso sobre la Asociación de Amigos del MSX. Dicha asociación, esta



formada por un grupo de usuarios y está destinada a la organización de reuniones entre otras cosas.

El siguiente paso lo dieron los chicos de TABUROTO, que presentaron su demo del COBRA MISION. Si mal no recuerdo se trata de un juego de PC, del cual han pasado las imagenes a MSX, ocupando estas 5 discos.

A continuación Elvis Gallegos nos presentó el VIDEO ANIMATOR, un programa de animación, que funciona a



USUARIOS DE MSX

base de digitalizaciones. Además, también dio a conocer la versión en inglés del RANDAR III.

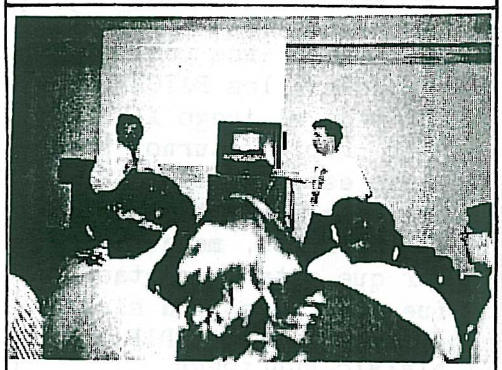
Y por fin llegó el momento esperado por la mayoría de los usuarios. Se dio a conocer que entre nosotros estaba un representante de COMPJOETANIA, que venía a presentar el PIXESS, un juego que debeis conocer por las dos demos que aparecieron en los SUNRISE PD. Patrick nos enseñó la demo del juego a la vez que explicaba su contenido y contestaba a las dudas, eso si, con la ayuda de Ramón Ribas que traducia del inglés al castellano.

Tras una larga demostración del PIXESS, le cedieron el turno a Jose Mª Alonso, quien nos ofreció una larga charla sobre NEXUS BBS y posteriormente sobre el emulador de MSX para PC.

Una vez aclarado lo anterior, Ramón Casillas realizo un sorteo con todos los productos de importación.

PYRAMID presentó su último PYRAMID MAGAZINE en disco, aunque no pudimos ver nada por estar comprimido. También nos enseñaron su demo de CINDY CRAWDFORD, de digitalizaciones en screen 12. Otro miembro de PYRAMID presentó otra demo, cuyo nombre es casi imposible de recordar. Estos chicos también ofrecerían más tarde unas demos de samplers, algunas para Music Module y otras para todos los chips.

Los siguientes fueron los responsables de TELEBASIC, que logicamente presentaron el TELEBASIC #2, que sigue la misma linea que el anterior, con algunos efec-



tos nuevos, como los vectores, y la novedad de contar con dos páginas en lugar de una sola como el anterior.

Acto seguido Ramón Casillas hizo su segunda aparición presentando el BABEL the new megablock, un juego de Daniel Zorita entre otros.

El siguiente fué Manuel Pazos, quien presento el SD-SNATCHER en castellano, y explicó que no solo está traducido, sino que también consta de algunas mejoras respecto a la version en inglés. El mismo presentó un disco con las músicas del Illusion City, y una pequeña demo para animar a los usuarios a que programen.

Hasta aquí llegó la primera parte de la reunión. Ahora debiamos ir a comer y luego volveriamos al cabo de unas horas. La espera se hizo corta, asi que alli estabamos de nuevo esperando sorprendernos de nuevo con alguna demostración.

El primero en coger el micro fue esta vez Ramón Ribas, quien nos demostró como funcinan los PATCHDISK, traduciendo el juego XAK II.

FKD repitio turno presentando esta vez el ALADIN, aunque, ya que todo debe decirse como es, me atrevo a opinar que esta presentación no fue tal, sino una simple comparacion del ALADIN con el DINAMIC PUBLISHER.

Otra vez cogió el micro Ramón Ribas para presentar otra producción de COMJOETANIA, el JUDGEMENT OF SOUND, una demo musical de alta calidad, y por supuesto en stereo.

Después de esto, Ramón Casillas presento una consola. SIN COMENTARIO.

Se dió tiempo para comprar, vender, hablar, probar cosas con el Turbo R,etc., y poco tiempo después se dio por concluida la reunión hasta Octubre, que se celebrará la siguiente.

Poco más podemos decir de la reunión, mas que asistais a la proxima. Os aseguro que no vais a salir decepcionados, ni mucho menos.

Para más información sobre la próxima reunión, os ponemos la dirección de Ramón Ribas, y os recordamos que él mismo anunció que los usuarios que quieran presentar algo deberan ponerse en contacto con la organización algunos dias antes.

A los que no pudisteis ir, y a los que lo hicisteis, os recordamos la existencia de un video rodado por nosotros, con el que os podreis enterar de lo que pasó. Os advertimos de nuevo que no somos SPILBERGs, por lo que no espereis un documental de la BBC, pero si quereis tener una idea más amplia de la reunión...



Ramón Serna

IMAGENES 3D

Todos debeis conocerla existencia de ciertos dibujos tridimensionales de los que la gente cuelga en sus escaparates y a primera vista parece la interpretación gráfica del sector de un disco. Me refiero a los que salen en el libro EL OJO MAGICO.

Para muchos de vosotros esto debe ser mágico, o por lo menos extraño. Sin embargo, Club Mesxes, no se deja intimidar por las cosas inexplicables, y ya tiene una ligera idea de como realizar dichos gráficos en un MSX.

Vamos a ocuparnos de aquellos en los que se ven diferentes superficies, es decir unos objetos se ven mas lejos que los otros.

Para realizar un gráfico en 3D lo primero es tener pequeños gráficos en 2D, por lo tanto es indispensable coger un editor grafico y realizar dos o tres dibujos de diferente tamaño a ser posible. El color y la forma da lo mismo.

Una vez hechos los dibujos, debeis copiar el primero de ellos en una fila horizontal, siempre a la misma distancia, hasta que llegueis a la otra punta de la pantalla. Acto seguido coged otro de los dibujos y repetid la operación, copiandolo un poco más abajo y a una distancia diferente. Seguid así hasta que lleneis la pantalla.

No importa si repetis los dibujos siempre que lo pongais a una distancia diferente de los demás. También es importante no dejar la pantalla con demasiados huecos entre dibujo y dibujo, pero lo más importante es que los dibujos de la misma fila horizontal deben estar todos a la misma distancia.

Una vez hecho esto, ya podeis empezar a mirar, sirviendos del reflejo del monitor si os es más cómodo.

Hay que notar que no todos los aparatos son buenos para esto, puesto que en algunos televisores, si son muy grandes cuesta un poco verlo. En monitores os aseguro que funciona perfectamente.

Por último, si no teneis la suficiente paciencia para hacer estos dibujos, os remitimos a la demo que ofrecerá Club Mesxes en la próxima reunión.

Ramón Serna

NOTICIAS V9990

Henrik Gilvad, y MI-Chi están trabajando en varios programas para el V9990, para que "usuarios normales" puedan usarla sin conocimientos de código máquina.

Mi-Chi ha hecho el FLI-player (Delta Animation player). El formato FLI no es compatible con el PC, pero ha sido optimizado para MSX. Tambien estan trabajando en un programa para ver con el V9990 ficheros *.GIF y *.BMP y en un programa de "video" para realizar toda una serie de efectos, aunque el propio Genrik nos ha dicho que no conoce muchos detalles sobre este último.

Henrik está adaptando el BASIC del TURBO R A1GT al V-9990. El BASIC del V9990 (POWER BASIC) tiene más comandos MSX y también mas screens, pero no son 100% compatibles poruqe no tiene SCREEN 0-4 (no exactamente compatible), y porque los sprites son mucho más avanzados que los de MSX-2. El BASIC de Henrik puede tener los siguientes screens:

SCREEN 0: modo de texto con 64 c/l de 8x8 pixels y definición multicolor.

SCREEN 1: --!!-Puede cambiarse a 32 c/l

SCREEN 2 y 4: 256x212 pixels con 16 colores (como screen 5). Tiene dos planos de gráficos, y cuando en el 1 se usa color 0 se puede ver el 2! Consta de 125 sprites multicolores (definidos como gráficos) de 16x16 pixels y con un máximo de 16 por linea sin parpadeos. Este modo gráfico consta de una tabla de caracteres, por lo que puedes hacer gráficos realmente rápidos como en la Sega Master System y la Megadrive!
Como el cartucho tien 512K de VRAM puedes definir 15.000 caracteres para ser seleccionados después . Los SCREENs 2-4 estan pensados para juegos!



SCREEN 3: 512x212 con 16 colores como en screen7, consta de 125 sprites y una tabla de caracteres, pero no tiene los dos planos.

SCREEN 5-12: Igual que en MSX, con solo 2 simples spri-

tes de 32x32 con 1 color o la operación lógica XOR.

SCREEN 5-7: Puede verse con un máximo de 1024x424 (entrelazado)

SCREEN 8-12: Maxímo de 512x426 (entrelazado)

Los Screens 10-12 pueden usar el sistema YUV en lugar del YJK. El YUV se usa más para imagenes comprimidas y funciona algo mejor que el YJK porque intercambia los componentes del verde y azul (G&B).

SET SCROLL: Funciona en todos los modos gráficos. En screen 2-4 tienes 4 parametros como estos:
SET SCROLL X1,Y1,X2,Y2
X1,Y1 es para el plano 1 y X2,Y2 es para el plano 2.
Los parametros 3º y 4º no son necesarios.

SET PAGE: Como el cartucho del V9990 tiene 512K de VRAM se multiplica por 4 el nº de páginas!

Muchos comandos gráficos son más rapidos que antes porque el V9990 utiliza dos ports de VRAM (DUAL PORT VRAM).
Con Screen 5-12 el VDP puede copiar a 2.2-2.5 Mb/seg incluyendo operadores lógicos (pset, xor, etc...).
La transferencia de CPU a VRAM es entre 500-550 Kb/seg. con T-R, mientras que con el V9958 tambien con T-R y en modo R-800 es de 120 kb/seg.
Lo peor del V9990, según Henrik, es que no tiene screen 0 compatible con MSX. El único modo de texto (rápido) como hemos dicho es de 64 c/l. Si alguien convierte el Driver ANSI de Uwe Schröeder a V9990 entonces se podran usar programas del DOS como Word Star a 80 columnas.
Henrik también tiene la adaptación del XBASIC a V9990. Este XBASIC es realmente asombroso si tienes un T-R! Especialmente en screen 2 o 4 pueden hacerce cosas muy buenas con los dos planos y los sprites. Como ejemplo, Henrik nos informa que Stefan Boer (autor del PUMPKIN ADVENTURE 2) tiene un V9990 y escribió una pequeña parte del P.A. 3 en su XBASIC en solo una tarde. El resultado fue tan rápido que tuvo que relentizarlo 4 o 5 veces (la version MSX del mismo programa está hecha en C.M. y sin relentizar).

© Henrik Gilvad / Ramón Serna



CD-ROM

Henrik También está escribiendo algunas rutinas y utilidades para poder leer y copiar ficheros de CD-ROM a HD o FD.

Algunos de sus programas pueden utilizar directamente los datos del CD-ROM pero normalmente, los demás programas del DOS no lo pueden hacer, porque todavia no ha hecho un "driver real" del DOS para el CD-ROM.

Uno de sus programas es para ver imagenes PCD (PHOTO CD). Este programa puede mostrar imagenes PCD en la V9990 en 3 modos diferentes de color y con 512x424 pixels. La mejor resolución es de 512x424 con colores reales RGB de 15 bits. El otro modo es el YUV y el screen 8. También está trabajando en una conversion para el YJK.

Para el TURBO R A1GT, ha hecho una rutina con la que pueden escucharse ficheros PCM (VOC, WAV, AIF, PCM, etc.) directamente mientras son cargadas del CD. Los ficheros pueden sonar sin interrupciones y sin tener que esperar mientras carga del CD. Esta rutina utiliza un bufer de 8-16 kb. para datos. Con el Music Module y el TURBO R, pueden sonar PCMs en stereo. La frecuencia puede llegar a 200 kbytes/seg ! (por tanto 22KHz y 44.1KHz no son del todo un problema!)

Hemnrik esta también trabajando en algunos comandos para el DOS (pequeños ficheros .COM para tener en el HD) para poder acceder al CD desde el DOS.

Ahora tiene el CDIR.COM para hacer el directorio principal, y va ha hacer el CCOPY.COM para copiar ficheros y el CCD.COM para cambiar de directorio.

En BASIC, con rutinas en C.M. está haciendo un programa para ver directorios y copiar ficheros del HD o FD al CD.

El CD-ROM es perfecto para el almacenage de grandes cantidades de ficheros MOD, PCM, GIF, BMP, etc. etc. etc. Pero puede que llegue a utilizarse el CD-ROM como un HD o un FD normal (sin poder escribir claro!).

■ Henrik Gilvad / Ramón Serna

OPL4

Sobre el OPL4 podemos decir que es un nuevo chip de YAMAHA, que tiene una parte FM igual a la del OPL3 (utilizada en la SOUNDBLASTER pro.) y es compatible con el OPL2 (ADLIB -PC-) y OPL1 (la parte FM del Music Module). El OPL3 tiene 18 canales FM stereo con 2 operadores por canal o 9 con 4 operadores por canal. Hay otras combinaciones, como 15 FM + 5 FM Drums (baterias). El OPL3 no es compatible con el OPLL (FM PAC) pero si lo es con el MUSIC MODULE. El OPL4 también tiene una parte PCM con la que puede hacer sonar 24 canales PCM stero a 44.1 KHz.

Henrik supone que los PCM normales tambien pueden sonar, pero aún no lo ha probado porque él no tiene el OPL4 aún. El solo tiene el OPL4 en una targeta de PC y en PC no puedes programar los chips PCM.

Por último, MOONSOFT está haciendo un programa para este nuevo chip, y Henrik ha hecho un driver para el PCM Tracker y un editor, aunque tan solo utiliza 9 canales FM + drums.

Henrik Gilvad / Ramón Serna

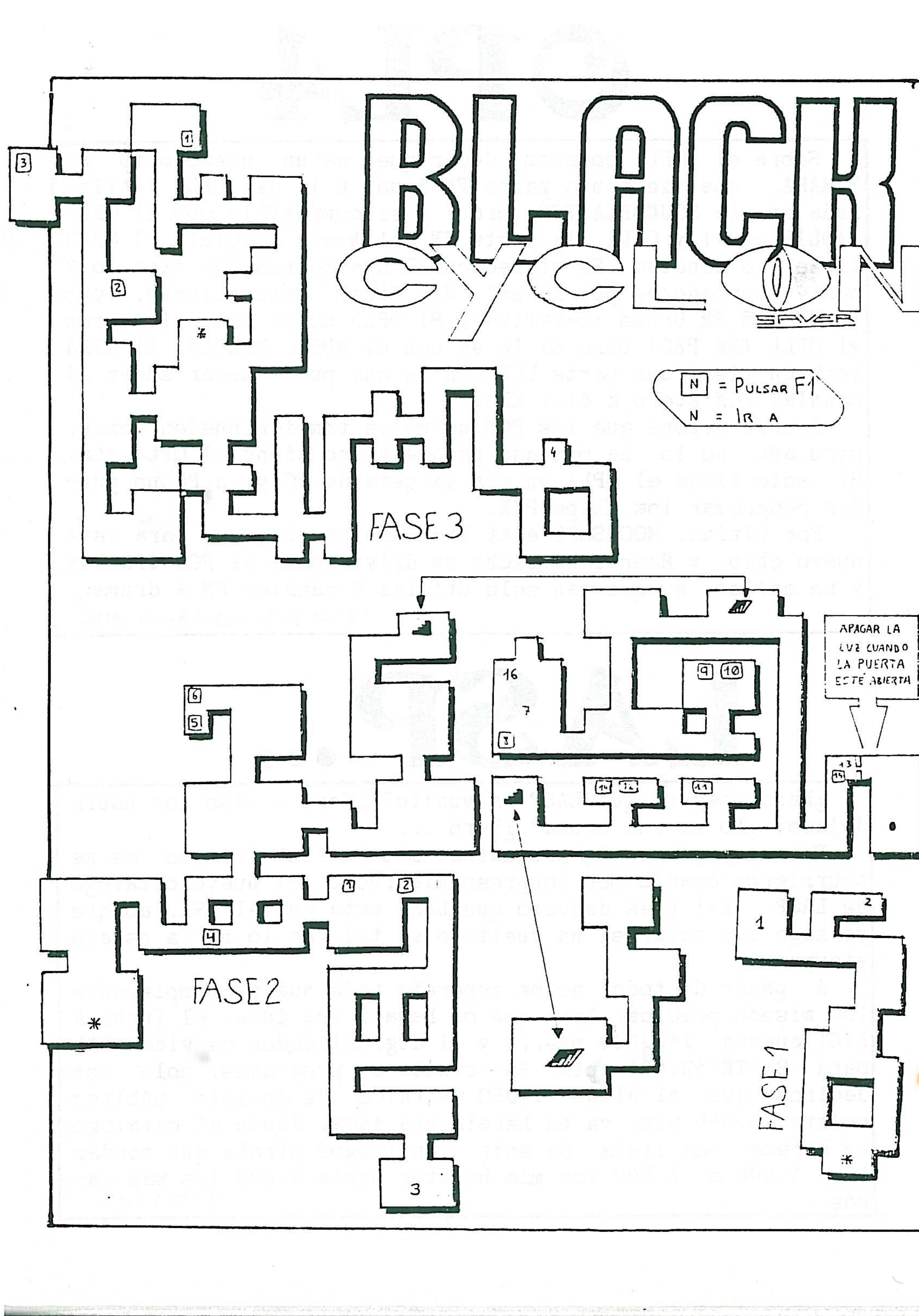
LASP...

¿Pero seguro que LASP ha vuelto? ¿Pero a caso nos había dejado? ¿No es una broma? ¿Pero ...

Estas y miles de preguntas más fueron las que se me ocurrieron cuando por sorpresa mia recibí el nuevo catalogo de LASP. Asi pues deduzco que LASP esta con el MSX, aunque yo sigo sin saber si ha vuelto o si tal vez lo había estado siempre.

A pesar de todo, no os espereis nada nuevo, simplemente los mismos precios. Para que os hagais una idea, el Turbo R A1GT cuesta 140.000 pts., y el digitalizador de video PAL para 2+/TR 52.000 pts. En cuanto a programas, solo con deciros que el FINAL VIDEO GRAPHICS de dominio público cuesta 10.000 pts. ya os hareis una idea. Junto al catalogo se ofrece una lista de soft con juegos pirata que rondan las 1.000 o 1.500 los más baratos hasta 4.000 los más caros.

Ramón Serna


BLACK CYCLON
SAVER
N = PULSAR F1
N = IR A
APAGAR LA LUZ CUANDO LA PUERTA ESTÉ ABIERTA
FASE 3
FASE 2
FASE 1

He aquí las 5 primeras fases de este explendido juego de PARALLAX. Podeis ver que los pasos ha seguir están enumerados, y que la fase 4 no ha sido incluida por tratarse de un enemigo final.

FASE 5

CALM DOWN, TRUROS.
MY NAME IS ERIC.
DO YOU KNOW WHY YOU
HAVE BEEN CAPTURED?

ENERGY
LIVES
LOCK OFF

Si habeis hecho el mapeado de algún juego no dudeis en mandarnoslo para que sea publicado en el próximo número.

J. Salvador Sanchez

SNATCHER™ プローグ



NUEVAS SECCIONES

En esta página, como veis, están 3 de las secciones más habituales en un fanzine: ANUNCIOS, TRUCOS Y POKES y OPINION. Si quereis participar en alguna de ellas,no dudeis en hacerlo escribiendo a la redacción.

Como en todo buen fanzine, he aquí nuestra sección de anuncios. Podeis enviar todos los anuncios de compra, venta, cambios, etc...No pondremos máximo en cuanto a palabras, siempre que no os paseis. De momento, como no tenemos anuncios vuestros, pondremos algunos de los propios miembros del club:

Vendo joystick YANJEN (el que se anunciaba en la MSX CLUB), funciona por tacto, y es muy preciso. Precio a convenir. O si no también lo cambio por otro periferico de interes. Enviad ofertas a la redacción.

Compro MSX-DOS 2.x, cartucho + discos, en buen estado. Pago hasta 4500 pts si está en perfecto estado. Negociables. Enviad ofertas a la redacción.

Intercambiamos todo tipo de soft para MSX-2, 2+ y Turbo R. Enviad lista o un disco para que os grabemos la nuestra a: la redacción.

¿ Os interesa saber todo lo que pasó en la V reunión de usuarios de MSX, en Barcelona día 8 de Mayo de 1.994 ? Pues ya podeis comprar el video que rodamos en cinta VHS por solo 995 pts. Más información: la redacción

Como es normal en cualquier fanzine, he aquí nuestra sección de "TRUCOS Y POKES". Para inagurarla, Manuel Pazos nos envia este truco:

-" Si en el menú del principio del PUMPKIN ADVENTURE II escribis:
WEWILLSTRIKEDOWNTHEONESWHOLEADUSWEAREYOURFUTUREWEAREFOREVER
(todo junto, sin espacios y en mayusculas) y luego pulsais SELECT, se producirá un flash. Al comenzar el juego lo hareis con todos los marcadores a tope."

Y la última sección, todavia sin estrenar, es la de "OPINION", o sea que si quereis opinar... ¡opinad!

Ramón Serna

ENSAMBLADOR

INTRODUCCION

Todos conocemos las limitaciones del BASIC en cuanto a velocidad y acceso pleno al hardware. Tambien sabeis todos que la solucion consiste en programar en ensamblador, pero esto puede llegar a resultar, digamos, un poco "dificil", dado que el lenguaje ensamblador no es muy amigable.

Existe una solucion intermedia consistente en realizar el esqueleto de nuestro programa en BASIC, y disponer en memoria de una coleccion de subrutinas en codigo maquina para realizar tareas especificas.

En efecto, esta es una solucion muy utilizada, ya que afortunadamente disponemos de los comandos DEFUSR y USR, con los cuales podemos invocar subrutinas en CM y devolver posteriormente el control al interprete BASIC.

Sin embargo, considero que esta no es la solucion ideal, por dos razones. La primera es que no es facil recordar ni identificar la funcion de una rutina en CM si nos limitamos a identificarla con un USRx: el programa pierde, por tanto, legibilidad. La segunda: con USR unicamente podemos pasar un parametro.

La solucion ideal seria crear tus propios comandos BASIC, darles el nombre que quieras y organizar el paso de parametros como mas te convenga. ¿Tres deseos irrealizables? No, gracias a CMD.

¿QUE ES CMD?

Simplemente es una palabra reservada del BASIC que no tiene asignada ninguna funcion.

Me explico. Cuando desde BASIC tecleamos un comando inexistente, el interprete salta a una rutina situada en ROM, encargada de generar un Syntax Error y devolver el control al modo directo. Si tecleas CMD obtendras Illegal Function Call. ¿Que ha pasado? Simple: cuando el interprete encuentra un CMD salta a la posicion &HFE0D. En condiciones normales simplemente encuentra un RET, por tanto vuelve y, esta vez, el basic si salta a la rutina de error.

Como muy astutamente habreis advertido, cuando trabajamos en BASIC &HFE0D se encuentra en RAM, por tanto nada nos impide realizar modificaciones. Tenemos 5 bytes disponibles: hacemos un POP BC para desactivar el salto a error, un JP que nos conducira a la rutina deseada y ...

CMDERO 1.0 por NESTOR SORIANO

El programa que presento para el manejo de CMDs permite dar nombre a tus subrutinas en CM y gestionar facilmente cuantos parametros quieras. Es decir, que el famoso

PRINT USRx(parametro)

se vera transformado en un mucho mas agradable

CMD nombre(param1,param2,...,paramn)

CMDero 1.0 consta de cuatro partes:

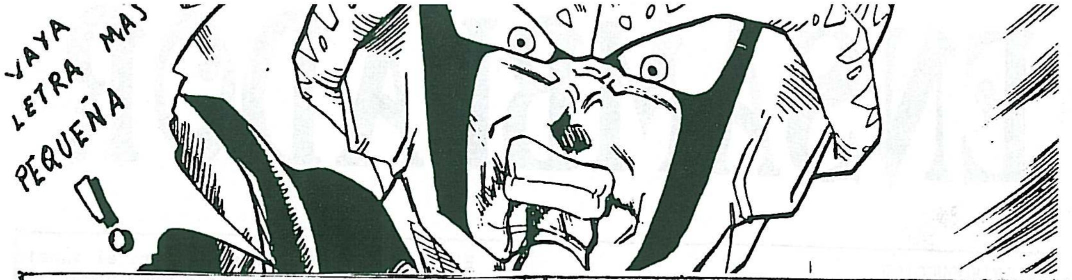
- Un programa que modifica el gancho de CMD.
- Un programa extractor y procesador del nombre del comando.
- Un paquete de rutinas de ayuda para la extraccion de parametros.
- Una zona de memoria en la que guardaras los nombres de tus comandos y la direccion de comienzo de los mismos.

La zona de nombres y las rutinas de extraccion de parametros son totalmente reubicables. La rutina de extraccion del comando precisa de algunas de las anteriores, por tanto su reubicacion implica unos pocos cambios que comentare mas adelante.

DESCRIPCION

A continuacion describo las rutinas que componen el CMDero; para una descripcion mas detallada de las rutinas del segundo bloque puedes mirar el propio listado.

- El programa modificador del gancho solo se ejecuta una vez por sesion; ocupa 16 bytes. En la posicion +&HE (offset respecto al comienzo) ha de figurar la posicion del programa extractor/procesador, y en +&H7 la posicion del propio modificador + &HB (todas las direcciones de memoria en formato byte bajo + byte alto).
- EXTRCOM extrae la cadena que figure tras el CMD (caracteres o TOKENS) y la almacena. Ocupa 28 bytes.
- COMPCAD compara dos cadenas. Ocupa 17 bytes.
- EXTRCAD se utiliza para la extraccion de parametros de cadena. Ocupa 28 bytes.
- FRMEVL extrae un parametro numerico. Se encuentra en ROM.
- PTRCHGET obtiene la direccion de almacenamiento de una variable de cadena. Ocupa 21 bytes.
- PTRGET actua como PTRCHGET pero para variables numericas. Se encuentra en ROM.
- MKERR genera en error. Esta en ROM.
- La rutina de extraccion y proceso extrae el comando existente tras el CMD (ex-



trae hasta encontrar un parentesis, dos puntos o un fin de linea). A continuacion lo compara con los nombres guardados en la zona de datos y, cuando encuentra una coincidencia, salta a la direccion asignada a ese comando. Ocupa 45 bytes.

En el caso de que queramos reubicar esta rutina o alguna de las anteriores debemos tener en cuenta lo siguiente:

* En la posicion +1 (respecto al comienzo de la rutina) hemos de colocar la ubicacion de BUFFER (posicion de almacenamiento temporal del comando extraido).

* En +4 hemos de colocar la ubicacion de EXTRCOM.

* En +9, la posicion de ALMACEN (zona de nombres de comando, ver mas adelante).

* Y en +12, la ubicacion de COMPCAD.

- La zona de almacenamiento de nombres de comando. Ha de tener el siguiente formato: * Primero, nombre del comando, finalizado por un signo "$". Solo garantizo el perfecto reconocimiento del mismo si empleas caracteres alfabeticos; prueba a experimentar con simbolos y numeros, quiza algunos funcionen. Hay que tener cuidado con dos cosas. La primera es que a la hora de escribir el comando desde BASIC da igual si lo haces en mayusculas o minusculas, pero en la zona de almacenamiento han de estar SIEMPRE EN MAYUSCULAS. Y ahi va la segunda: si el nombre del comando contiene una palabra reservada del BASIC no hay que almacenar esta palabra como tal, sino como su token correspondiente (un token es un byte que el BASIC asigna a cada instruccion con el fin de ahorrar memoria).

Por ejemplo, si queremos un comando llamado NESTOR debemos almacenar los codigos ascii de "N","E" y "S"; a continuacion &HD9, que es el token de "OR"; despues el codigo de la "R" y finalmente el "$".

A continuacion un procedimiento facil para averiguar el token correspondiente a una instruccion: desde BASIC realiza un programa de una unica linea, que contendra unicamente la instruccion a investigar. En la posicion &H8005 encontraras el token correspondiente.

Ojo, algunos tokens constan de dos bytes; los reconoceras porque el primero de ellos es &HFF. Examina entonces &H8006 para obtener el segundo.

Con un ejemplo comprobaras que no es tan dificil. Supongamos que queremos almacenar un comando llamado EXPRESION. Vemos que contiene dos palabras reservadas: EXP y ON. Hacemos el programa:

```
10 EXP
```

y vemos que &H8005 contiene &HFF. Por tanto examinamos &H8006, y encontramos &H8B. A continuacion hacemos

```
10 ON
```

y en &H8005 encontramos &H95. Ya podemos almacenar el nombre del comando:

```
FF 8B "R" "E" "S" "I" 95 "$"
```

* Tras el nombre del comando se ubica su direccion de comienzo (byte bajo + byte alto), seguido de un signo "*".

* Repetimos para todos los comandos. Al final de la lista colocaremos un caracter cualquiera seguido del signo "#".

- El programa necesita una zona para almacenar el comando invocado y poder compararlo. La posicion de esta zona ha de ser indicada en el segundo y tercer byte de la rutina extractora/procesadora.

AHORA TE TOCA A TI ...

- Ya solo quedan tus propios programas, que pueden extraer parametros con ayuda de las rutinas del segundo bloque.

Cuando el CMDero pasa el control a tu programa, el puntero de la ejecucion del programa BASIC se encuentra en la pila: no olvides ir dejarlo listo para la siguiente instruccion cuando tu programa acabe; de lo contrario generaras un Syntax Error. Si el comando tiene parametros, dicho puntero esta situado en el parentesis inicial; si no, esta en la siguiente instruccion. Ten en cuenta que tras cada extraccion el puntero queda en el separador de parametros (recomiendo usar una coma) o en el parentesis final, por lo que habra que incrementarlo cada vez. En base a todo lo anterior, un programa invocado con el CMDero que no use parametros debe tener la forma siguiente:

```
ORG direccion de inicio
... (rutina con igual numero de PUSH
que de POP) ...
POP HL ; Recuperamos puntero BASIC
RET
```

Y uno que use parametros quedara como:

```
ORG direccion de inicio
POP HL
INC HL ; Pasamos el parentesis
... (Extraccion y proceso del primer
```

parametro) ...

```
    INC HL ; Pasamos la coma
    ... (Extraccion y proceso del segundo
  parametro) ...
    INC HL
    (Asi para todos los parametros) ...
    INC HL ; Pasamos el parentesis final
    RET
```

Recuerda que las rutinas extractoras de parametros actualizan automaticamente el puntero BASIC, dejandolo en el siguiente separador. Por descontado, has de guardar el puntero cada vez que necesites usar el par HL.

Al final del listado hay un ejemplo de comando con extraccion de parametros, llamado NESTOR; la zona de almacenamiento esta preparada para reconocerlo y ejecutarlo.

PASO DE VARIABLES

Hasta ahora hemos visto como pasar parametros para lectura. Estos parametros podian ser cualquier expresion que diera como resultado un numero o una cadena. Queda pendiente el paso de variables para su modificacion.

Lo ideal seria poder usar el CMD como funcion; es decir, poder hacer variable = CMD nombre (parametros)

No he conseguido hacer esto (si tu lo consigues escribeme contandome como lo has hecho), pero si podemos hacer que un parametro resulte modificado (por supuesto, ha de ser una variable). Para ello usa las rutinas PTRGET y PTRCHGET del segundo bloque. El comando SORIANO del final del listado es un ejemplo de modificacion de una variable entera.

REUBICACION

He intentado hacer esta rutina lo mas reubicable posible. Las rutinas de extraccion se pueden instalar donde quieras sin problemas; la rutina modificadora del gancho y la extractora/procesadora precisan de los cambios indicados en DESCRIPCION.

Si tus propias rutinas no emplean parametros de cadena solo necesitaras llamar a la ROM para extraer parametros, por tanto puedes reubicar el CMDero sin modificarlas. Pero si tus rutinas emplean EXTRCAD, COMPCAD o PTRCHGET deberas actualizarlas cada vez que reubiques el CMDERO.

El final de la zona de usuario (o lo que es lo mismo, el principio de la zona de trabajo del sistema) cambia segun las unidades de disco. Si no tienes, este limite se situa siempre en &HF380. En caso contrario, examina las posiciones &HF378 y &HF379 para conocer el comienzo de la zona sagrada.

Existe una posicion de la zona de trabajo usada como buffer de la sentencia PLAY; comienza en &HF975 y tiene una longitud de 384 bytes. Puedes colocar aqui el CMDero SOLO si estas seguro de que NO VAS A USAR PLAY NI UNA SOLA VEZ. Te recomiendo redactar un programa instalador que permita reubicar el CMDero facilmente, realizando automaticamente los cambios pertinentes en el modificador del gancho y en el extractor/procesador. Yo he realizado uno, pero es demasiado extenso para publicarlo; si lo quieres mandame un disco y te lo grabare.

CMDERO COMO CONJUNTO

Hasta ahora he presentado el CMDero como un conjunto de rutinas independientes, pero lo mas logico es considerarlo como un solo programa y guardarlo como tal en memoria; comprobaras que es asi como figura en el listado. En estas condiciones, CMDero (sin incluir la zona de nombres de comando) ocupa 140 bytes, y la posicion de las diferentes rutinas, respecto al principio del conjunto, es la siguiente:

EXTRCOM: +0
COMPCAD: +&H1C
EXTRCAD: +&H2D
PTRCHGET: +&H49
Extractor/procesador: +&H5E (recuerda que, a su vez, esta rutina ha de contener la ubicacion de BUFFER, EXTRCOM, ALMACEN y COMPCAD)
ALMACEN: +&H8C

FORMATOS DE ALMACENAMIENTO

A continuacion describo como se almacenan los parametros extraidos en el DAC (acumulador decimal):

- Las variables enteras se almacenan en &HF7F8, byte bajo + byte alto. VALTYP (&HF663) contiene un 2.
- Las variables de simple precision en &HF7F6, un byte de exponente (el primer bit es el signo) y tres de mantisa (cada byte contiene dos cifras en formato BCD). VALTYP contiene un 4.
- Las variables de doble precision se almacenan como las de simple, pero hay siete bytes destinados a la mantisa. VALTYP contiene un 8. - Si extraemos una cadena, VALTYP contiene un 3; recuerda que hay una rutina especifica para extraer cadenas en el segundo bloque.

El formato de almacenamiento de las variables es la siguiente: Un byte que indica el tipo (2,3, 4 u 8); dos bytes para el nombre y 2, 4 u 8 bytes para almacenar la variable, en el mismo formato que el DAC. Cuando llamas a PTRGET no obtienes la direccion de la variable, sino la del dato contenido. Por tanto, para averiguar el formato de una variable deberas llamar a PTRGET, restar 3 a DE y examinar (DE).

CONCLUSION

Hace poco que estoy metido con esto del ensamblador, por tanto seguro que esta rutina puede ser ampliamente optimizada. No dudes en escribirme para cualquier sugerencia, duda, critica, comentario, autografo, carta-bomba, etc... La direccion esta en la pagina de este vuestro fanzine, pon en el sobre "para la seccion de KONAMI MAN"

Gracias/Perdon/Hasta la proxima.

```
1 : //////////  CMDERO BAI NESTOR SORIANO. JULIO & AGOSTO 1.994 \\\\\\\\\\
2 :
10 HCMD:      EQU  &HFE0D
20 MKERR:     EQU  &H4072
30 CHRGTR:    EQU  &H4666
40 FRMEVL:    EQU  &H4C64
50 FRESTR:    EQU  &H67D0
60 PTRGET:    EQU  &H5EA4
70 NEWSTT:    EQU  &H4601
80 DAC:       EQU  &HF7F6
90 VALTYP:    EQU  &HF663
100 ;
110 ;####### PRIMER BLOQUE. MODIFICACION DEL GANCHO HCMD (&HFE0D)  ######
120 ;
130           ORG  &HB000
140           LD   BC.5
150           LD   DE.HCMD
160           LD   HL.HDAT
170           LDIR
180           RET
190 ;-- Nuevo CMD --
200 HDAT:     POP  BC
210           JP   PROGRAMA
220           NOP
230 ;
240 ;##### SEGUNDO BLOQUE: RUTINAS DE EXTRACCION DE PARAMETROS  #######
250 ;
260           ORG  &HD000
270 ;
280 ; -- EXTRCOM: Extraccion del comando --
290 ; -- (Extrae hasta encontrar un parentesis abierto.
300 ; --          o hasta encontrar otra instruccion o un fin de linea)
310 ; -- Entrada: HL = puntero del BASIC justo despues de CMD
320 ; --          DE = direccion para colocar el comando
330 ; -- Salida:  DE = direccion de inicio de la cedena de comando
340 ; --          (al final hay un signo "$")
350 ; --          HL = puntero del basic en la siguiente instruccion,
360 ; --          o bien justo despues del parentesis.
370 ; --          o bien se ha vuelto al modo directo.
380 ; --          BC = Longitud del comando sin el "$"
390 ;
400 EXTRCOM:  PUSH DE
410           LD   BC.0
420           DEC  HL
430 BUCLE:    CALL CHRGTR
440           JR   Z.FIN
450           CP   "("
460           JR   Z.FIN
470           CP   0
480           JR   Z,FIN
490           LD   (DE).A
500           INC  DE
510           INC  BC
520           JR   BUCLE
530 FIN:      LD   A,"$"
540           LD   (DE).A
550           POP  DE
560           RET
570 ;
580 ;-- COMPCAD: Rutina de comparacion de cadenas --
590 ;-- Entrada: Hl y DE apuntan a las dos cadenas a comparar
600 ;-- Salida: A = 1 si las dos cadenas son iguales: 0 si no
610 ;
620 COMPCAD:  LD   A.(DE)
630           CPI
640           JR   NZ.NO
650           CP   "$"
660           JR   Z.SI
670           INC  DE
680           JR   COMPCAD
690 NO:       XOR  A
700           RET
710 SI:       LD   A.1
720           RET
730 ;
740 ;-- EXTRCAD: Extraccion de un parametro de cadena --
750 ;-- Entrada: HL = Puntero al principio del parametro
760 ;--          (situado automaticamente por EXTRCOM
770 ;--           o por una anterior ejecucion de EXTRCAD)
780 ;-- Salida:  A = Longitud de la cadena
790 ;--          VALTYP (&HF663)<>3 si el parametro no es una cadena
800 ;--          DE = Puntero al inicio de la cadena
810 ;--          HL = Puntero al siguiente parametro o al parentesis
                                                               \ final
820 ;
830 EXTRCAD:  CALL FRMEVL
840           LD   A.(VALTYP)
850           CP   3
860           RET  NZ                 ; Si no es una cadena no extrae
870           PUSH HL
880           CALL FRESTR
890           EX   DE.HL
900           POP  HL
910           PUSH DE
920           POP  IX
930           LD   A.(IX+0)
940           LD   E.(IX+1)
950           LD   D.(IX+2)
960           RET
970 ;
980 ;-- FRMEVL : Extrae un parametro --
990 ;-- Entrada: HL = Puntero al parametro (igual que EXTRCAD)
1000 ;-- Salida:  A = 2. 4 u 8 segun el tipo de valor adquirido
1010 ;--          VALTYP = 3 si el parametro es una cadena
1020 ;--          (para extraer cadenas hay que usar EXTRCAD)
1030 ;--          DAC (&HF7F6) = Parametro extraido
1040 ;--          HL = Puntero al siguiente parametro,
1050 ;--          o al parentesis final.
1060 ;
1070 ;-- FRMEVL se encuentra en ROM: &H4C64
1080 ;
1090 ;-- PTRCHGET: Extraccion de la dirrecion de almacenamiento
1100 ;--           de una variable de cadena
1110 ;-- Entrada: HL = Puntero al nombre de la variable en el texto
1120 ;-- Salida:  HL = Puntero BASIC despues de la variable
1130 ;--          BC = Direccion de almacenamiento de la longitud de la
1140 ;--          DE = Direccion de almacenamiento de la cadena | cadena
1150 ;--          A<>3 si la variable no es de tipo cadena
1160 ;
1170 PTRCHGET: CALL PTRGET
1180           PUSH HL
1190           EX   DE.HL
1200           DEC  HL
1210           DEC  HL
1220           DEC  HL
1230           LD   A.(HL)
1240           INC  HL
1250           INC  HL
1260           INC  HL
1270           PUSH HL                ; Direccion de la longitud
1280           INC  HL
1290           LD   C.(HL)
1300           INC  HL
1310           LD   B.(HL)
1320           PUSH BC                ; Direccion de la cadena
1330           EX   DE.HL
1340           POP  DE
1350           POP  BC
```

Metal gear mk-II 1988.

```
1360 ;
1370 ;-- PTRGET: Extraccion de la direccion de almacenamiento de una variable
1380 ;--         (Para variables de cadena hay que usar PTRCHGET)
1390 ;-- Entrada: HL = Direccion del nombre de la variable en el texto BASIC
1400 ;--          SUBFLG (&HF6A5) = 0 si la variable es simple,
1410 ;--                          <>0 si es una matriz
1420 ;-- Salida:  HL = Puntero BASIC despues del nombre de la variable
1430 ;--          DE = Direccion de almacenamiento de la variable
1440 ;--          DE-3 = Direccion de almacenamiento del tipo de variable
1450 ;--
1460 ;-- PTRGET se encuentra en ROM: &H5EA4
1470 ;
1480 ;-- MKERR: Generacion de error --
1490 ;-- Entrada: E = Numero de error a generar
1500 ;
1510 ;-- MKERR se encuentra en ROM: &H4072
1520 ;
1530 ;******* TERCER BLOQUE: RECONOCIMIENTO DEL COMANDO
1540 ;        Y SALTO A SU DIRECCION ASIGNADA *******
1550 ;
1560 PROGRAMA: LD   DE,BUFFER
1570           CALL EXTRCOM
1580           PUSH HL              ;Puntero BASIC => pila
1590           PUSH DE              ;Puntero al comando extraido
1600           LD   HL,ALMACEN
1610 COMPARAR: CALL COMPCAD
1620           OR   A
1630           JR   NZ,YEAH
1640 PAOH:     POP  DE
1650           PUSH DE
1660 SIGUIENT: INC  HL
1670           LD   A,(HL)
1680           CP   "*"
1690           JR   Z,FINISH
1700           CP   "*"
1710           JR   NZ,SIGUIENT
1720           INC  HL
1730           JR   COMPARAR
1740 YEAH:     EX   DE,HL
1750           LD   A,(DE)
1760           LD   L,A
1770           INC  DE
1780           LD   A,(DE)
1790           LD   H,A
1800           POP  DE
1810           JP   (HL)
1820 FINISH:   LD   E,18
1830           JP   MKERR
1840           NOP
1850 ;
1860 ;******* CUARTO BLOQUE: ALMACEN DE NOMBRES DE COMANDO
1870 :        Y SUS DIRECCIONES DE EJECUCION *******
1880 ;
1890 ALMACEN:  DEFM "NES"              : Ver seccion de ejemplos
1900           DEFB &HD9,"R","$"       : &HD9 = Token de TO
1910           DEFW &HC000
1920           DEFB "*"
1930           DEFB "S",&HF7           : &HF7 = Token de OR
1940           DEFM "IANO$"
1950           DEFW &HC040
1960           DEFB "*"
1970           DEFM "RSC$"
1974 ; Ha nacido Nestor !
1980           DEFW &HF980
1990           DEFB "*"
2000           DEFB "X","0"
2010 BUFFER:   DEFS 50                 : Almacen del comando extraido,
                                         usado para comparacion)
2020 :
2030 :******* EJEMPLOS *******
2040 :
2050 :
2060 :-- Ejemplo de lectura de parametros --
2070 ;-- Uso: CMD NESTOR(Entero,Cadena)
2080 ;-- Salida: &HC200 y &HC201 guardan el entero
2090 ;--         &HC202 guarda la longitud de la cadena
2100 ;--         A partir de &HC203 esta la cadena
2110 ;
2120 EJPANU:   ORG  &HC000         ; Ejemplo de extraccion de un parametro
2130           POP  HL             : Recuperamos puntero BASIC       numerico
2140           INC  HL             : Pasamos del parentesis
2150           CALL FRMEVL
2160           LD   A,(&HF663)
2170           CP   2              : Es un parametro tipo INTEGER?
2180           JR   Z,BIEN
2190           LD   E,13
2200           JP   MKERR          : Si no, Type mismatch
2210 BIEN:     LD   BC,(&HF7F8)
2220           LD   (&HC200),BC    ; Almacena numero en &HC200 (2 bytes)
2230           INC  HL             ; Siguiente parametro
2240 EJPACA:   CALL EXTRCAD        : Ejemplo de extraccion de un parametro
                                                                de cadena
2250           PUSH AF
2260           LD   A,(&HF663)
2270           CP   3              : Es un parametro de cadena?
2280           JR   Z,BIEN2
2290           POP  AF
2300           LD   E,51
2310           JP   MKERR          : Si no, Internal error (Je, je!)
2320 BIEN2:    POP  AF
2330           LD   (&HC202),A     ; Almacena longitud de cadena en &HC202
2340 ALMACENA: PUSH AF             : Almacena variable a partir de &HC203
2350           POP  BC
2360           PUSH HL             : El puntero BASIC, bien guardadito
2370           LD   HL,&HC203
2380           EX   DE,HL
2390           LDIR
2400           POP  HL             : Recuperamos el puntero BASIC y...
2410           INC  HL
2420           RET                 : ...continuamos con el programa.
2430 ;
2440 ;-- Ejemplo de modificacion de una variable --
2450 ;-- Uso: CMD SORIANO (Variable entera)
2460 ;-- Salida: Deposita el valor 65 en al variable pasada
2470 ;
2480           ORG  &HC040
2490           POP  HL
2500           INC  HL
2510           CALL &H5EA4
2520           LD   A,65
2530           LD   (DE),A
2540           XOR  A
2550           INC  DE
2560           LD   (DE),A
2570           INC  HL
2580           RET
```

FIN

▶メタル・ギア mk-II (SR以降)

SOFT NACIONAL

Esta sección esta dedicada a comentar software de producción española, ya sean juegos, demos, o utilidades. La razón es bastante simple, necesitamos que alguien patrocine el soft nacional, que es mucho mejor de lo que la mayoría cree. Prueba de ello, son los muchos programas que se presentaron en la V Reunión de Usuarios de MSX, que a pesar de no estar todos terminados, prometian ser bastante buenos.

Para que os comentemos el programa solo teneis que enviarnos una copia de este, aunque no esté acabado e incluso si quereis podemos distribuirlo a un precio razonable.

Para empezar esta sección comentaremos un juego que será presentado en la próxima Reunión de Usuarios, en Octubre.

Nombre: MESTRE FIGHTER
Casa: D'OLI CLAR GAMES
Tipo: Lucha
Formato: 1DD
Chip: R-800/Z-80
Lenguaje: BASIC/KUNBASIC
Música: FM+PCM
Graficos: Scr 7



El juego sigue la misma linea de los conocidos juegos de lucha, como el STREET FIGHTER. Los graficos están muy bien cuidados, utilizando decorados y personajes de la serie DRAGON BALL. A pesar de estar muy bien presentado, falla en la velocidad, puesto que esta completamente en BASIC. Las músicas, hechas con el PCM TRACKER, contarán con baterias en PCM, siempre y cuando su autor consiga el DRIVER para Turbo R de este programa.

En conclusión, para ser el primer programa que va a presentar al publico D'OLI CLAR GAMES no se puede pedir mucho más. En cuanto esté listo, os informaremos más detalladamente.

■ Ramón Serna / J. Salvador Sánchez

PROGRAMAR Y PROGRAMAR ... UNA OBSESION

Este artículo, realizado por Enrique Martinez, es la introducción de lo que serán varios cursillos de programación, en BASIC y ENSAMBLADOR.

No podemos contar solo con nuestra inventiva e ingenio, hay que aplicar al proceso de programar una serie de disciplinas que nos permitan un desarrollo rápido y eficaz.

Ante un problema, lo primero que debe hacerse es idear una solución, de la manera más ordenada posible y, en la medida de lo posible, bien documentada, es decir, adjuntar toda la información posible para saber bien lo que se está haciendo, o para que otros programadores puedan realizar modificaciones o mejoras a nuestros programas.

Para programar es necesario tener un ordenador y un lenguaje que nos permita una comunicación con la máquina, pero hay "lenguajes" y "lenguajes". Aunque parezca mentira, el BASIC es un lenguaje de alto nivel.

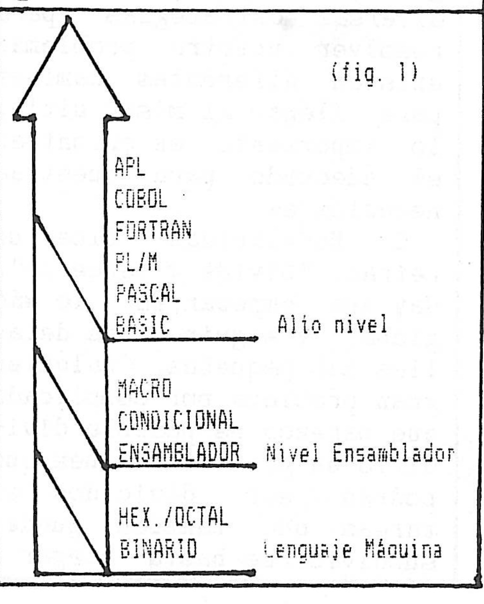
LENGUAJES DE ALTO NIVEL

Los lenguajes de alto nivel constan de instrucciones potentes que hacen la labor de programación mucho más rápida y fácil. Por lo general, a cada instrucción de alto nivel corresponden muchas binarias. Para que os hagais una idea de los lenguajes de programación: (fig. 1)

Sea cual fuere el lenguaje elegido para programar, lo mejor es ponerse manos a la obra y empezar con la programación siguiendo unos ...

CONSEJOS

1. ¡No perder la calma! Ante cualquier problema que surja, a grandes problemas grandes soluciones.



(fig. 1)

2. Definir el problema, que vamos a solucionar, con pelos y señales (cuanta más información se tenga, mejor) una serie de preguntas que nos pueden resurgir en nuestro problema pueden ser:

a) ¿Gráficos? ¿Sprites?
b) ¿Registros de RAM/VRAM?
c) ¿Rapidez? ¿Es necesaria?

3. Documentación. Hay que documentarse, lo primero, desde el principio, antes de sentarse ante el teclado. La realización de la documentación es un ejercicio para detectar ambigüedades de nuestro problema (programa), inconsistencias de definiciones y otros fallos dificiles de solucionar más tarde. La documentación ha de ser: legible, completa y revisada y si es diaria mejor todavía.

4. Pensar y después programar. Hay que estudiar diversas estrategias para resolver nuestro problema, existen diferentes caminos para llegar al mismo sitio, lo importante es encontrar el adecuado para nuestras necesidades.

5. Modularidad. Dice un refrán: "Divide y vencerás". Hay que empezar por lo más global y seguir a los detalles más pequeños. Cualquier gran problema por complicado que parezca es posible dividirlo en grandes bloques que podrán ser divididos en tareas que tambien pueden subdividirse hasta llegar a procesos sumamente pequeños y sin dificultad.

6. Pensar en los demás. Es necesario que la organización general de los distintos modulos sea comprensible para los demás. Por ello es imprescindible disponer las distintas partes de forma aislada y dejando claro su objetivo. Es importante que estos módulos o partes no sean grandes ni complejas.

7. Desarrollar módulos. Cada modulo o parte del programa debe estar perfectamente definido según sus entradas y salidas. La forma en la que se desarrollan los cálculos internamente no interesan a los demás módulos. Esto convierte a los módulos desarrollados en reutilizables en diferentes situaciones, dentro de la misma aplicación o en otras distintas (como las subrutinas).

Esta forma de programar nos permite una mejora, ya que para acelerar los programas solo tenemos que "retocar" unos módulos (los más lentos) y no todo el programa.

8. Hacer las constantes, constantes. Siempre es mejor incluir las constantes al principio del programa y en un módulo que sólo sea para almacenar las constantes. De esta forma ganaremos tiempo y sabremos en todo momento las constantes que tenemos.

Enrique Martinez

CONCLUSION

Esto ha sido todo por ahora, creo que para ser el primer nº no ha estado del todo mal, os hemos ofrecido una amplia sección de noticias, hemos resumido la V Reunión de Usuarios, sin olvidarnos de los programas en BASIC y ENSAMBLADOR. También os hemos comentado el S.PD.#12, cosa que haremos en cada nº (¡con los siguientes S.PD., claro!).

Y por supuesto, muchas más cosas que sin duda recordareis. Como veis nos las hemos apañado bastante bién, asi que imaginaos lo que haremos cuando contemos con vuestra colaboración. De momento tenemos varias secciones a la espera de vuestras cartas, que son las de anuncios, trucos, opiniones, soft nacional, y por supuesto todo aquello que querais comentar, ya sean juegos, cualquier utilidad o algún programa de "producción casera".

Nuestras puertas están abiertas a todas las sugerencias, al igual que aceptaremos agradecidos cualquier critica, sea positiva o negativa, ¡aunque por supuesto preferimos las primeras!

Todos los que esten dispuestos a formar parte de nuestro grupo de colaboradores recibirán un ejemplar gratuito, pero antes deberán ganárselo.

Ahora ya profundizando un poco, espero que el fanzine os haya gustado, y sobre todo que lo hayais comprado. Digo esto porque se por experiencia que para un fanzine lo más importante es que lo compren. Si el fanzine se vende, la moral de los autores se mantiene alta y cada vez se vuelve más y más interesante. Sin embargo, si el fanzine no se vende sus autores se unden, y el fanzine se va a pique.

Por nuestra parte, promocionaremos nuestro fanzine todo lo que podamos, a fin de que la gente se interese por verlo y lo compre.

¡Si la cosa funciona tendreis fanzine para rato!

Ramón Serna

Club Mesxes
C/ Manacor 16, 1º 1ª
07006 - Palma de Mallorca
Baleares



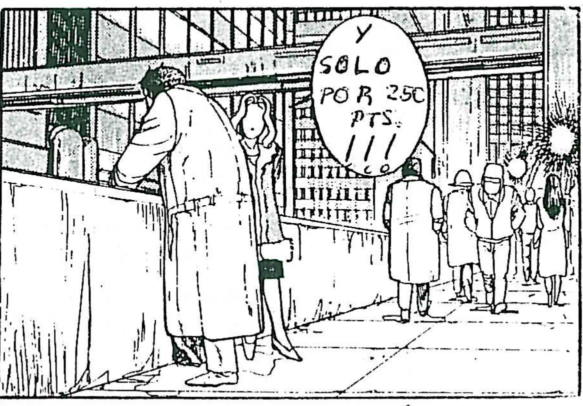
D MESXE




¿HABEIS OIDO LO DEL PRO-XIMO Nº?
SÍ, HABLA-RAN DEL SPD #13!


¡DICEN QUE HABRÁ MÁS NOTICIAS!
¡Y MÁS CMDs!


¡COMENTA-RÁN EL BLADE LORDS!
¡¡¡LO ÚLTIMO DE PARALLAX!!!


¡¡¡Y SOLO POR 250 PTS.!!!